

WHAT IS CLAIMED:

1. A method of designing a process lifecycle using a computer system comprising:
presenting a series of user interfaces allowing a process architect to define a process lifecycle
using business model objects as building blocks; and
5 presenting input indications in said series of user interfaces allowing a process architect to
specify what parts of said defined process lifecycle can be deleted or modified; and
wherein said parts of said process lifecycle comprises one or more of said business model
objects or one or more relationships between said business model objects.
2. A method of initiating a product development process using a computer system
10 comprising:
presenting one or more user interfaces allowing a program manager to select from one or more
defined process lifecycles;
presenting a series of user interfaces allowing a program manager to modified those parts of a
selected process lifecycle that are specified as modifiable in said process lifecycle; and
15 presenting a series of user interfaces allowing a program manager to make assignments to roles
in said process lifecycle.
3. A method of executing a product development program using a computer system
comprising:
using an instance of a product development process, with one or more predefined roles
20 assigned to one or more process implementers, to coordinate activity of various resources;
presenting one or more user interfaces to one or more process implementers to provide a task
list to said one or more process implementers;
presenting one or more user interfaces to one or more process implementers to receive data
indicating completed or incompletd tasks from said one or more process implementers;
25 aggregating data received from said one or more process implementers into project summary
data; and
presenting project summary data to a program manager.
4. The method of claim 1 wherein one or more business objects are associated with one
or more states that characterize a business object's status.
- 30 5. The method of claim 4 wherein business objects when first created have a particular
state.
6. The method of claim 4 wherein behavior of a business object depends on one or more
business rules.

7. The method of claim 1 wherein business objects can transition between states as a result of changes of state of other business objects in accordance with one or more business rules.

8. The method of claim 7 wherein business rules can be defined during the initial design of a lifecycle or during the modification of a lifecycle for a particular project and can also be imposed by the overall design of the software system.

9. The method of claim 4 wherein business objects can be combined to form a structure or hierarchy where rules associated with a business object are based on one or more factors comprising:

- contents of said business object;
- business rules of said business object;
- relationships of said business object to other business objects;
- parent business objects of said business object.

10. The method of claim 4 wherein business objects can be combined to form a structure or hierarchy where rules associated with a business object are based on one or more factors comprising:

- contents of said business object such as a gate review business object cannot be complete until all its questionnaires are complete; a project business object cannot be completed until all its phases are complete;
- business rules of said business object such as lifecycle applicability rules that determine for which type of program a lifecycle can be used;
- relationships of said business object to other business objects such as when a deliverable has a start to finish relationships with another deliverable said deliverable cannot go active until a predecessor deliverable is complete; and
- parent business objects of said business object such as a phase is a parent to a deliverable, a lifecycle is a parent to phases, a methodology is a parent to lifecycles and as further examples, a workflow process in a deliverable is automatically initiated when the deliverable becomes active; when a phase is activated, the deliverables it contains are activated depending said deliverables relationships.

11. The method of claim 2 wherein said computer system presents interfaces to a program manager through which said manager:

- inputs profile information;
- receives and reviews candidate lifecycles;
- selects a desired lifecycle;
- modifies a selected lifecycle;

creates an instance of a selected and/or modified lifecycle for a particular development program; and
assigns users to predefined roles for said particular development program.

12. The method of claim 1 wherein said business model objects can comprise one or more of: methodology, lifecycle, role, phase, deliverable, resource assignment, fixed cost, and risk.

13. A computer system software engine usable for designing process lifecycles and managing and executing instances of process lifecycles for particular projects comprising: one or more methodologies;
10 wherein each of said methodologies comprises one or more similar lifecycles;
wherein each of said lifecycles comprises one or more phases; and
wherein each of said phases can comprise one or more deliverables.

14. The system of claim 13 further comprising:
wherein each of said lifecycles can comprise one or more of role, cost, resource, and risk data.

15. The system of claim 13 further comprising:
wherein each of said phases can comprise one or more of role, cost, resource, and risk data.

16. The system of claim 13 further comprising:
wherein each of said deliverables can comprise one or more of role, cost, resource, and risk data and one or more workflows, templates, and forms.

17. The method of claim 1 wherein said states can comprise one or more of: pending, planning, active, complete, inactive, canceled; and additional states.

18. The method of claim 1 wherein object state transitions can be manual or automatic.

19. The method of claim 1 wherein automatic object state transitions occur based on similar transitions of other related objects.

20. The method of claim 1 wherein an object state transition can cause other cascading object state transitions that thereby automate aspects of the development process.

21. The method of claim 1 :
wherein a resource assignment object can be initialized to be activated just-in-time, e.g., only after all predecessors are complete;
30 wherein activation of a resource assignment object triggers one or more task notifications; and
further comprising:

automatically notifying a process implementer using said computer system of a task in response to said activated resource assignment object.

22. A method of managing a product development process using a computer system comprising:

5 defining elements of a process lifecycle in a structured hierarchy of phases and deliverables;
wherein once said structured hierarchy of phases and deliverables is specified, said computer system is capable of enforcing required aspects of said process lifecycle;
wherein once said structured hierarchy of phases and deliverables is specified said computer system automates execution of a program by distributing assignments as they are needed and providing a continuously updated living schedule integrating progress status of all aspects of a program.

10 defining states associated with phases and deliverables that characterize their status;
after said defining, providing access to one or more process managers to input initial information regarding phases and deliverables including relationships and dependencies
15 between phases and deliverables and state goals for phases and deliverables;
providing access to said computer system to one or more process implementers in order for said implementers to enter data indicating changing status of phases/deliverables;
in accordance with said defined process/lifecycle phases and deliverables, informing one or more process implementers of updated lifecycle resource needs and due dates;
20 in response to a request from a manager, providing overview and drill-down reports of updated process/lifecycle status.

23. The method of claim 22 wherein said elements of a process/lifecycle comprise schedules, tasks, relationships, documents and resource requirements.

24. The method of claim 22 wherein said elements of a process/lifecycle comprise hourly cost, required skills, and competency levels.

25. The method of claim 22 further comprising:
allowing a process architect to indicate what parts of the process/lifecycle are mandatory and what parts (where parts comprise objects, their relationships and the lifecycle itself) can be changed by a program manager or team in order to enforce process parameters.

30 26. The method of claim 22 further comprising:
allowing a process architect to classify a lifecycle based on a series of user-defined criteria that will determine the conditions under which the lifecycle can be used, wherein said user-defined criteria can comprise one or more of:
the type of product being developed;

the market to which the product will be sold; and
a business unit for which the product is intended.

27. The method of claim 26 further comprising assisting a program manager in selecting the most appropriate lifecycles for a development program.

5 28. The method of claim 22 further comprising:
allowing a program manager to indicate other users that will part of a program by assigning
individuals to roles specified in a program lifecycle;
creating an association between roles or users and and program tasks and
thereby supporting automated execution (putting things on user tasks list, communicating with
10 users, etc.) when the program is activated.

29. The method of claim 28 further comprising:
sending tasks to a user's personal task lists when said tasks are needed, based on approvals and
the progress of work on related tasks or groups of tasks.

30. The method of claim 29 wherein a task sent to users may have linked to them
15 documents or other information needed to complete said task.

31. The method of claim 28 further comprising:
providing one or more users (both process implementators and project managers)_with real
time/living schedule reports that reflect the latest updates and revisions.

32. The method of claim 31 further wherein said updates and revisions comprise
20 revisions made by users to their tasks via a communication channel.

33. The method of claim 31 further comprising:
wherein said updates and revisions comprise addition or removal of tasks or groups of tasks
from the overall process via a communication channel.

34. The method of claim 22 further comprising:
25 enforcing a consistent process structure/hierarchy (lifecycle, phase, deliverable, resource);
enforcing a consistent mapping of organizational structure (divisions, business units);
consolidating schedule, cost, risk and resource information; and
providing a user with a requested report at any requested level in the process hierarchy and for
any requested part of the company.

30 35. The method of claim 22 further comprising:
comparing real time forecast data from individual users to plan values for schedule and costs;
changing the state of an indicator when user defined tolerances are exceeded; and

notifying users of impending schedule slips or cost overruns.

36. The method of claim 35 further wherein notification of a slip in a schedule is escalated to higher level reports in the process hierarchy only when said slip occurs on a schedule critical path, thereby making potential schedule delays along the critical path visible in the highest level reports.

37. The method of claim 36 further comprising allowing a user to view an alert of a slip in a higher level report and allowing the user to drill down to more detailed, lower level reports to get to the source of said slip.

38. A method of evaluating and comparing a group of product development programs in a portfolio using a computer system comprising:

allowing a user to define program-specific metrics that will be tracked by said computer system;

allowing a user to define how metric values will be obtained during execution of a program; and

presenting to a user multi-program portfolio data regarding multiple programs' phase, cost, schedule, and risk status.

39. The method of claim 38 further wherein said metrics can be derived from system data (e.g. cost to date).

40. The method of claim 38 further wherein said metrics can be derived from user input during reviews (e.g. sales forecasts).

41. The method of claim 38 further wherein said metrics can be derived from quantitative responses to one or more questions

42. The method of claim 38 further wherein said metrics can be derived from a user-defined mathematical formula involving one or more other metrics (e.g. metric 1 divided by metric 2).

43. The method of claim 41 further comprising:
for metrics derived from questionnaire responses, allowing a user to define the questions to be associated with the metric; and
for metrics derived from questionnaire responses, allowing a user to define a questions response scale.

44. The method of claim 41 further comprising:

allowing a user to specify which users will receive an electronic questionnaire that will be used to capture responses.

45. The method of claim 41 further comprising:

allowing users to analyze and discuss user responses, and to enter a consensus score to be used
5 in calculating metric values for program and multi-program analysis.

46. The method of claim 38 further comprising providing users with metric reports to support program review and portfolio level decision making, said metric reports derived from one or more of:

system data;
10 user input during reviews;
quantitative responses to one or more questions;
user define mathematical formula involving one or more other metrics.

47. The method of claim 38 further comprising:

allowing users to compare program attractiveness and performance by creating customized
15 tabular reports and charts of the programs and metrics they wish to analyze.

48. A network-based method for automating requesting and assigning resources to work on projects comprising:

allowing a user to search for available resources/users with the skill and competency level
required to accomplish tasks on a development project with a single action;
20 returning to a user a list of resources/users;
including in said returned list an analysis of how a proposed assignment will impact overall
utilization of indicated resources;
including in said returned list an analysis of how well the resources are able to satisfy the
demands of the assignment;
25 allowing the user to request a single user or group of users from one or more users acting as
resource managers, via the web;
routing a request to the appropriate users acting as functional managers for review;
providing reports that show the detailed impact of the assignment on the requested user(s) and
allows the user acting as the functional manager to approve, reject, or propose an
30 alternative user; and
routing the functional managers decision back to the requesting user for review, who can
accept the decision or make another resource request, wherein accepting the decisions
automatically assigns the user in question to the program and gives the assigned user him
access to the web based program workspace.